Solving the Captive Portal Problem

NSLondon Q4

Ross Butler

@ross_w_butler

Captive Portals

"A captive portal is a web page which is displayed to newly connected user before they are granted broader access to network resources."

- Wikipedia

Problem

- Wi-Fi hotspots with captive portals present as though unsecured
- iOS connects automatically to known networks
- Generic error handling can lead to poor reviews

Reachability

Used by iOS developers as the de facto means of checking whether an Internet connection is available.

"A remote host is considered reachable when a data packet, sent by an application into the network stack, can leave the local device. Reachability does not guarantee that the data packet will actually be received by the host."

- Apple Developer Documentation

WISPr

- Wireless Internet Service Provider roaming (WISPr) protocol published by the Wireless Broadband Alliance
- The specification defines the Universal Access Method (UAM)
- Captive Network Assistant (CNA) software component (Smart Client)

CNA

Contacts a number of known URLs hosting a small webpage:

e.g. http://captive.apple.com/hotspot-detect.html

```
<HTML><HEAD><TITLE>Success</TITLE></
HEAD><BODY>Success</BODY></HTML>
```

If the content cannot be retrieved then a captive portal is blocking the connection. iOS can present the web page for user interaction to access the network.

User-Agent: `CaptiveNetworkSupport-390.0.3 wispr`

Network Framework

- `NWPathMonitor` used to monitor changes in network state

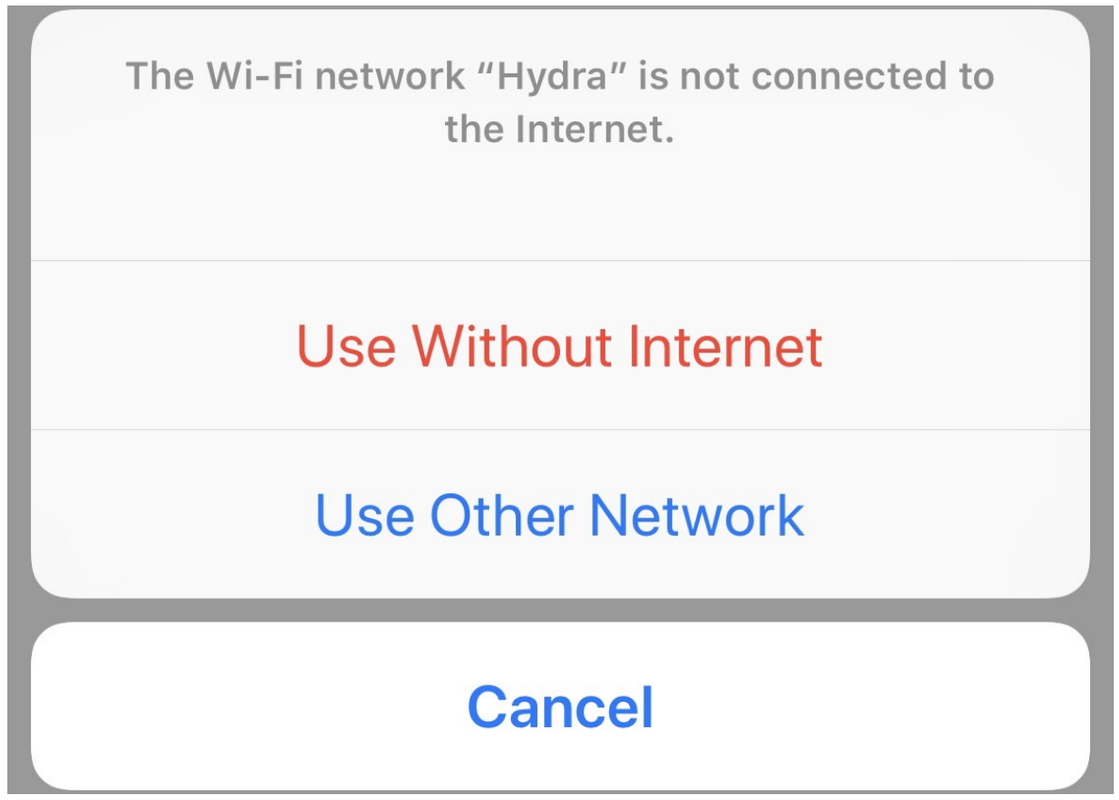
- `NWPath` (satisfied, unsatisfied & requiresConnection)

```
let monitor = NWPathMonitor()
let monitor = NWPathMonitor(requiredInterfaceType: .wifi)
monitor.pathUpdateHandler = { path in
    if path.status == .satisfied {
        print("Connected")
    }
}
let queue = DispatchQueue.global(qos: .background)
monitor.start(queue: queue)
```

Network Framework

- `NWPathMonitor` reports the path as being `satisfied` after negotiating the captive portal

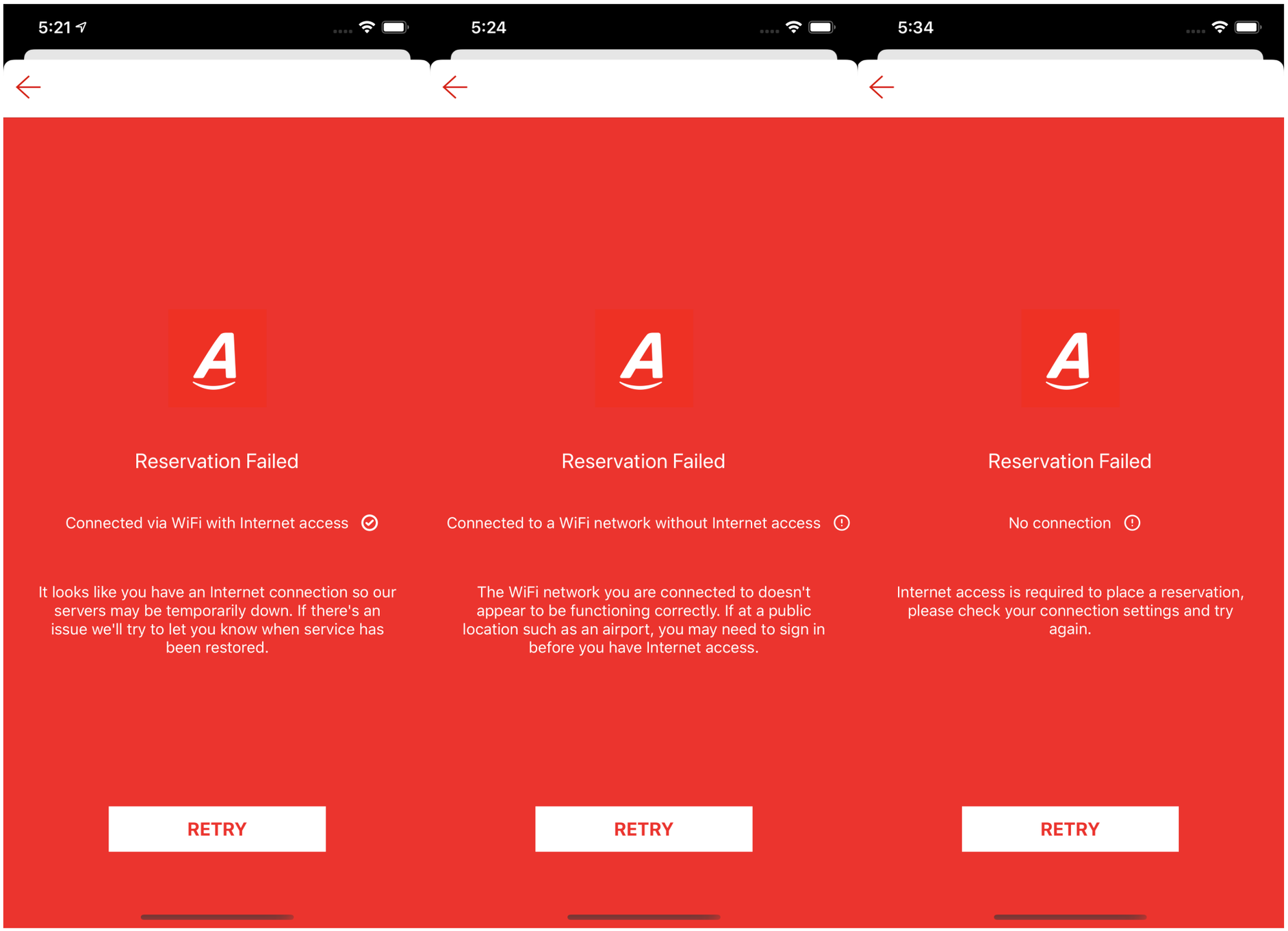
- Where there is no captive portal to present to the user an action sheet is presented offering the options *Use Without Internet* or *Use Other Network*.



Connectivity

- Provides a familiar API similar to Reachability.
- Emulates the behaviour of iOS by contacting a set of URLs and checking the response where cellular or Wi-Fi available.
- Connectivity URLs & percentage of successful connectivity checks may be specified.
- Polling option available for situations where a persistent connection is required.

Diagnostics



API

```
let connectivity: Connectivity = Connectivity()

connectivity.connectivityURLs =  [URL(string: "https://
www.apple.com/library/test/success.html")!]

let connectivityChanged: (Connectivity) -> Void = { [weak
self] connectivity in
     self?.updateConnectionStatus(connectivity.status)
}

connectivity.whenConnected = connectivityChanged
connectivity.whenDisconnected = connectivityChanged
connectivity.startNotifier()
connectivity.stopNotifier()
```

API

```
func updateConnectionStatus(_ status: Connectivity.Status) {

  switch status {
      case .connectedViaWiFi:
      case .connectedViaWiFiWithoutInternet:
      case .connectedViaCellular:
      case .connectedViaCellularWithoutInternet:
      case .notConnected:
  }


}
```

Connectivity

Cocoapods

```
pod “Connectivity"
```

Carthage

```
github “rwbutler/Connectivity”
```

Swift Package Manager

https://github.com/rwbutler/Connectivity

References

- Captive Portal - Wikipedia
  - https://en.wikipedia.org/wiki/Captive_portal
- Detecting Internet Access on iOS 12+
  - https://medium.com/@rwbutler/nwpathmonitor-the-new-reachability-de101a5a8835
- Reachability
  - https://developer.apple.com/library/archive/samplecode/Reachability/Introduction/Intro.html
- SCNetworkReachability - Apple Developer Documentation
  - https://developer.apple.com/documentation/systemconfiguration/scnetworkreachability-g7d
- Solving the Captive Portal Problem on iOS
  - https://medium.com/@rwbutler/solving-the-captive-portal-problem-on-ios-9a53ba2b381e